Computers Math. Applic. Vol. 17, No. 11, pp. 1467–1473, 1989
Printed in Great Britain.

0097-4943/89 \$3.00 + 0.00

A TUTORIAL ON THE EXTENDED PERTURBATION ANALYSIS

S. Li

Department of Systems and Industrial Engineering, The University of Arizona, Tucson, AZ 85721, U.S.A.

(*Received* 27 *June* 1988)

Abstract—In this paper, we explain the simple idea of infinitesimal and extended perturbation analysis, the motivation of the new technique, the difficulties, as well as the advantages and contributions.

1. INTRODUCTION

A new challenge to control community is the classes of man-made systems known as the *Discrete Event Dynamical Systems* (DEDS; Levis *et al.* [1]). The need for analysis, optimization, and control of this kind of systems is rapidly increasing due to the fast development of modern computer networks, communication networks, as well as automated manufacturing systems. The difficulties of applying the rich results of existing control and system theories are mainly that of lack of analytically tractable modelling techniques which satisfy the necessary system design and computational requirements as identified in Ho [2]. This is in contrast to the long developed differential/difference equation model for aerospace and mechanical systems. Different models exist and new models are being created, such as the existing queueing model, Petri net model, the new supervisory control model [3], communicating sequential process model [4], min–max algebra model [5], the generalized semi-Markov process model [6] etc. However, there is still no dominating model for DEDS. To meet the demand of the modern society, a lot of more efforts must be made.

Perturbation analysis (PA) among other new developments is a technique in the area of discrete event dynamical systems. One of the first problem which triggered the research on this new technique is a real production line problem [7]. It was found that the problem can be reduced to the abstract form of a serial of queues with finite buffers as shown in Fig. 1. The system contains n machines $M_1, \ldots, M_n$, with finite buffers $B_1, \ldots, B_n$ in front of each machine. The question asked is then how to optimally allocate the buffer sizes in terms of, say, system throughput of the line. This problem is an open problem for a long time, and in fact there is no general computationally tractable exact solutions. Existing method for dealing with this kind of problem as the theory of queueing network requires basic assumption like exponential service times at each machine, infinite buffer sizes, independent service times etc. [8]. There is also method using the finite state Markov chain theory, but as the number of machine and buffer sizes become big, the computational efforts needed is out the reach of the fastest modern computers. One universal method is of course the discrete event simulation techniques. The disadvantage of simulation is its inflexibility. For example, to design the buffer sizes in the above system, we need to answer n "what if" question such as "what if the ith buffer size increase by 1?". Clearly, for each of these questions, we need to change the parameter of the buffer and simulate the new perturbed system. Thus to answer the n "what if" questions, we need $n + 1$ simulation, where the additional one simulation is the one for the nominal system. This method of brute force simulation is of course very time consuming and sometimes impossible.

Perturbation analysis strives to solve this problem. By observation of only the nominal simulation or an observation of the sample path of the real system, PA tries to answer all the n "what if" questions without running the additional n simulations for each of the perturbed buffer sizes. The method was accidentally started by solving the real production line problem. It was then realized that this is actually a very general idea and can be applied to general DEDS [9]. Excitements, difficulties and controversies have been on all the way of PA from the very early stage. A lot of progress has been made [10, 11], among which is the so called extended perturbation

Fig. 1. A production line with finite buffers.

analysis [12, 13]. In this paper, we explain using a very simple example the idea of infinitesimal perturbation analysis (IPA) and extended perturbation analysis (EPA).

We now recapitalize and generalize the above discussion. We represent a discrete event dynamical system (to be more specific later) by a stochastic process $x_\theta(t, \omega)$† parametrized by a parameter θ. A performance measure of the system is given as a functional of the process $L(x_\theta(t, \omega))$. Our goal is then to optimize the expected performance measure $J(\theta) = E(L(x_\theta(t, \omega))$. We also assume that the problem cannot be solved analytically and thus the optimization is based on simulation or observation of the real process. Thus we may use, say, stochastic approximation algorithms of the type [14]

$$\theta_{n+1} = \theta_n + a_n h(g_n, \theta_n),$$

where $g_n = \{[J(\theta_n + \Delta\theta)]_{\text{est}} - [J(\theta_n)]_{\text{est}}\}/\Delta\theta$ is the gradient estimation in the simulation at the point θ_n, and a_n, h can be chosen according to the particular algorithm used. The question is then reduced to the estimation of the gradient g_n for a given set point θ_n. Hence we use perturbation analysis technique.

The paper is organized as follows: in Section 2, we introduce a very simple system for the paper. We discuss the basic concepts and ideas of the IPA. We then present the difficulties of IPA and motivate EPA. In Section 3, we give the new idea of extended perturbation analysis. Finally, in Section 4, we briefly discuss the relation to the acceptance and rejection technique of simulation, as well as the recent new algorithms inspired by EPA. The paper ends with a concluding remark.

2. INFINITESIMAL PERTURBATION ANALYSIS AND ITS DIFFICULTIES

Let us consider a very simple system throughout this paper. The system is artificially designed to illustrate the basic ideas of PA and properties of DEDS. It represents by no means the complication of the general DEDS. Consider a two state Markov process with states 0 and 1. The transition probability of the imbedded Markov chain is: $p^{00} = \theta$, $p^{01} = 1 - \theta$, $p^{10} = p^{11} = 0.5$, where p^{ij} denotes the probability of going to state j upon leaving state i. The holding time distribution function at state x is: $F_x(t) = 1 - e^{-\lambda(x,\theta)}$, for $x = 0, 1$, where λ is some known function to be specified later. The transition rate diagram of this Markov process is shown in Fig. 2.

Let the performance measure be $L(\theta, \omega) = T_{00}$ = steady state time between successive visits of $x = 0$. We are interested in the parametric optimization problem $\min_{\theta \in [a,b]} E\{L(\theta, \omega)\}$, with the assumption that we choose to use simulation.

A sample path of this process is shown in Fig. 3a denoted as the nominal path (i.e. for the nominal parameter θ). Now we show how we can generate a sample path of this process in simulation. To generate a sample path, we need to generate both the holding times according to $F_x(t)$ and the routings according to p_{ij}.

Rule (*i*). Generating the state holding times: let u be a uniformly distributed random variable‡ in [0, 1]. Let $S_x(\theta, \omega)$ be a sample state holding time at state x. We can generate this time by $S_x(\theta, \omega) = F_x^{-1}(u) = -[1/\lambda(x, \theta)]\ln(1 - u)$. It is easy to prove that the $S_x(\theta, \omega)$ so generated has distributed as F_x [15].

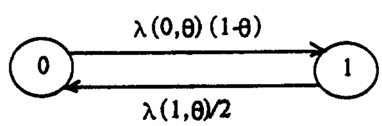

Fig. 2. State transition rate diagram.

†$\omega \in \Omega$, where (Ω, F, P) denotes the underlying probability space for the random process.
‡This is available in simulation.

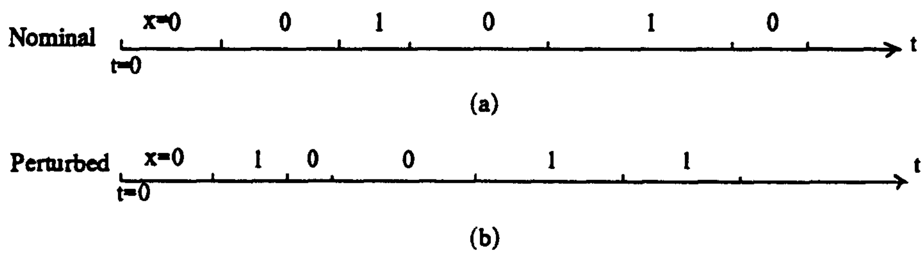

Fig. 3

Rule (ii). Generating the routings: let v be a uniformly distributed random variable in [0, 1], and $r_x(\theta, \omega)$ be the sample routing random variable taking values in $\{0, 1\}$. We can generate this by

$$r_x(\theta, \omega) = \begin{cases} 0 & \text{if } x = 0 \text{ and } 0 \leqslant v < \theta, \text{ or } x = 1 \text{ and } 0 \leqslant v < 0.5, \\ 1 & \text{otherwise.} \end{cases}$$

Again, it is easy to prove that $r_x(\theta,\omega)$ has the right distribution.

With the above two random variable generation mechanisms, we can generate two such sequences of i.i.d uniformly distributed random variables $\{u_k\}_{k \geqslant 1}$, and $\{v_k\}_{k \geqslant 1}$ and form a sample path of the process $x_\theta(t, \omega)$ according to the above rules.

Now remember that our purpose is to get the sensitivity of the performance (i.e. g_n), thus given a nominal system parametrized by θ, we need to consider a slightly perturbed system with parameter $\theta' = \theta + \Delta\theta$. Let us try to do the variational analysis along a simulated or observed nominal sample path (e.g. the one in Fig. 3a). The effects of the perturbation $\theta \to \theta + \Delta\theta$ on the sample path are:

Case (i). Holding time perturbation: $S_x(\theta, \omega) \to S_x(\theta, \omega) + \Delta S_x$, where from rule (i)

$$\Delta S_x(\theta, \omega) \approx \frac{\mathrm{d}S_x(\theta, \omega)}{\mathrm{d}\theta} \Delta\theta = \frac{\mathrm{d}\lambda(x, \theta)/\mathrm{d}\theta}{\lambda^2(x, \theta)} \ln(1 - u)\, \Delta\theta = -\frac{\mathrm{d}\lambda(x, \theta)/\mathrm{d}\theta}{\lambda(x, \theta)} S_x\, \Delta\theta. \tag{1}$$

Case (ii). State sequence change: let the current states on the nominal and the perturbed paths be x, then the next state on the perturbed path will be the same as that of nominal one if either $x = 1$ (in which case the routing distribution p_{li} does not depend on θ) or $\mathbf{1}_{[0,\theta]}(v) = \mathbf{1}_{[0,\theta+\Delta\theta]}(v)$† [in which case the perturbation on θ does not affect the routing according to rule (ii) above]. If this condition is violated, namely if at state $x = 0$ and $\theta + \Delta\theta \leqslant v < \theta$ (we assume that $\Delta\theta < 0$), then the next states on the nominal and perturbed paths will be different by rule (ii). In this case, on the nominal path, the next state will be 0, but on the perturbed path, the next state will be 1 instead. Fig. 3b shows a perturbed path.

To motivate the extended perturbation analysis, we apply the earliest version of PA, the infinitesimal perturbation analysis (IPA) to our system. A key simplifying assumption made by the infinitesimal perturbation analysis is the following: for the purpose of estimating derivative $\mathrm{d}L/\mathrm{d}\theta$, we can choose $\Delta\theta$ arbitrarily small, such that we assume that at state $x = 0$ on the nominal path, if we find $0 \leqslant v < \theta$, then $v < \theta + \Delta\theta$ (note we have chosen $\Delta\theta < 0$). Thus case (ii) above never occurs, and *the state sequences on the nominal and the perturbed paths are identical.* The only effect of perturbation $\Delta\theta$ is on each state holding time. The assumption that the two state sequences are identical is the so called *Deterministic Similarity* (DS) assumption, which is assumed by IPA. With this DS assumption, the sample path derivative can be very easily calculated as follows: for any typical $T_{00}(\theta, \omega) = S_0^1 + S_1^1 + \cdots + S_0^{n(\omega)}$, where $n(\omega)$ is the number of visits to state $x = 1$ before visiting state $x = 0$, starting from $x = 0$, and S_i^k is the kth holding time at state i. Now by equation (1) and the deterministic similarity assumption, the IPA estimate of ΔT_{00} is

$$\Delta T_{00} = \Delta S_0^1 + \sum_{k=1}^{n(\omega)} \Delta S_1^k = -\frac{\mathrm{d}\lambda(0, \theta)/\mathrm{d}\theta}{\lambda(0, \theta)} S_0^1\, \Delta\theta - \sum_{k=1}^{n(\omega)} \frac{\mathrm{d}\lambda(1, \theta)/\mathrm{d}\theta}{\lambda(1, \theta)} S_1^k\, \Delta\theta. \tag{2}$$

†Here the function "1" denotes the indicator function.

Note that there is no perturbation in $n(\omega)$ because of the DS assumption. From (2), we have

$$\frac{dT_{00}(\theta,\omega)}{d\theta} = -\left\{\frac{d\lambda(0,\theta)/d\theta}{\lambda(0,\theta)} S_0^1 + \sum_{k=1}^{n(\omega)} \frac{d\lambda(1,\theta)/d\theta}{\lambda(1,\theta)} S_1^k\right\}. \tag{3}$$

Hence, by IPA, the estimate in (3) converges to $E\{dT_{00}(\theta,\omega)/d\theta\} = E\{dL(\theta,\omega)/d\theta\}$, which is the expected value of the sample path derivative. Clearly, the IPA estimate $dT_{00}/d\theta$ in (3) can be easily recursively calculated as we observe the evolution of the nominal path.

The IPA rules are simple to perform, and no additional simulation runs for the perturbed paths are needed. Thus it is very efficient. Now the question is that of whether

$$E\{dL(\theta,\omega)/d\theta\} = dE\{L(\theta,\omega)\}/d\theta, \tag{4}$$

where the right- and left-hand sides are the estimate we want and the estimate IPA offers, respectively. In another word, we must prove the validity of DS assumption. It has been known [16] that the DS assumption will be violated in most of the cases. But under certain conditions the IPA estimates are still strongly consistent [17, 18]. However, there are still other systems like multiple class queueing systems, priority queues etc., which cannot be solved by IPA as shown by Cao [19] and Heidelberg *et al.* [20].

Now let us have a look at our example to discover the difficulties of IPA. We compute both sides of (4) as follows: the stationary probability of state 0 can be computed from balance equation according to the transition rate diagram Fig. 1 as $\pi(0) = \lambda(1,\theta)/[\lambda(1,\theta) + 2(1-\theta)\lambda(0,\theta)]$, and hence

$$J(\theta) = E\{T_{00}\} = E\{S_0(\theta,\omega)\}/\pi(0) = \frac{\lambda(1,\theta)/\lambda(0,\theta)}{\lambda(1,\theta) + 2(1-\theta)\lambda(0,\theta)}. \tag{5}$$

Therefore, the right-hand side of (4) can be obtained by simply taking derivative on (5). On the other hand, the left-hand side of (4) can be obtained by taking the expectation of (3). Now in order to show that the two sides are not consistent, let us choose a particular form of the rate $\lambda(x,\theta)$, i.e. $\lambda(0,\theta) = \lambda(1,\theta) = 1$. In such case, from (5), $dJ(\theta)/d\theta = dE\{T_{00}\}/d\theta = 2/[1 + 2(1-\theta)] > 0$, for all $\theta \in [0, 1)$. But from (3), $E\{dT_{00}/d\theta\} = 0$. Hence, IPA estimate is not correct in this case. Figure 4 gives typical sample path of $T_{00}(\theta,\omega)$ for a fixed ω and $E\{T_{00}(\theta,\omega)\}$ as functions of θ. Clearly, the jumps in $L(\theta,\omega)$ contribute to the average performance measure $J(\theta)$. Thus we need to relax the DS assumption. But one can see from Fig. 3 that once NP and PP begin to differ, the two paths will be very different thereafter. How can one infer one from the other if they are very different?

To summarize the insights we get from our example, we have seen the following: the sample performance measure $L(\theta,\omega)$ of DEDS is usually discontinuous. This is caused by the occurrence of state sequence changes of perturbed path from that of nominal path. In order to tackle this problem, we need to relax the DS assumption. But once DS is relaxed, the nominal path and the perturbed path may differ very much so long as states differ. Thus allowing viol-

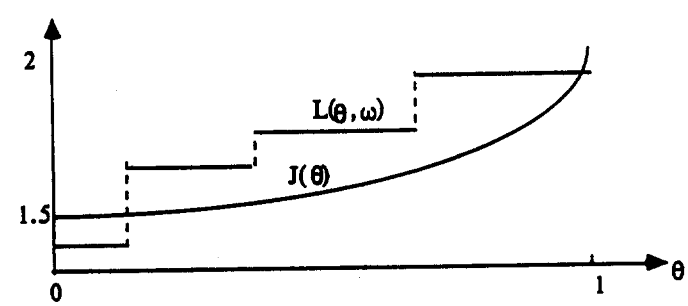

Fig. 4

ation of DS condition makes the inference of perturbed information from nominal one very difficult.

3. EXTENDED PERTURBATION ANALYSIS

In this section, we present the basic idea of the extended perturbation analysis (EPA) technique, still within the framework of our example. EPA presents one way of solving the dilemma discussed in the last section. We first give the following very simply property of Markov process.

Consider a homogeneous ergodic Markov process $x(t)$. Given any state x_1, let S be any time when the process is in state x_1. Then there exists a random time (stopping time) T such that the process visits x_1 after S. Let us define a new process $y(t)$ as the following:

$$y(t) = \begin{cases} x(t) & \text{for} \quad 0 \leqslant t < S, \\ x(t+T) & \text{for} \quad S \leqslant t. \end{cases}$$

The process $x(t)$ and the definition of $y(t)$ is depicted in Fig. 5.

Intuitively, by the strong Markov property, the process $y(t)$ so defined obey the same random law as $x(t)$ (for a detailed proof, the reader is referred to Li [21]). The intuitive interpretation of this result is the following: We can *cut* one piece of the process $x(t)$ and *paste* onto another piece to form a new sample path without changing the statistical property, provided that we guarantee the same joint state x_1. We thus refer this property as the *invariance under cut-and-paste* property.

Now we show how to make use of this simple property of Markov process in our construction of a perturbed process. We use our cut-and-paste idea to construct a perturbed path which is not necessarily the one generated by brute force simulation. Actually, any path which is a sample path of the perturbed process is sufficient for the sake of sensitivity estimation.† Conceptually, we perturbed the nominal path as we did in Section 2 using IPA until we test that the next states will be different on the two sample path, i.e. whenever at state $x = 0$, $\theta + \Delta\theta \leqslant \omega \leqslant \theta$. For ease of explanation, we re-plot Fig. 3 in Fig. 6, adding on the top a sample path called the constructed perturbed path, which will be the one we constructed from the nominal path without an additional simulation.

In Fig. 6, we first give NP, we perturb the first state holding time according to (1). At the end of this state, we test whether $\theta + \Delta\theta \leqslant v \leqslant \theta$, which is "yes" in the figure. Since the 2nd perturbed state becomes 1 instead of 0 on NP, we can simply use the cut and paste idea as follows: we stop the construction of the CPP at time S until we find at a later time T, state 1. Then we cut the portion of NP from T and paste onto time $S + \Delta S$ on CPP. The perturbation on the holding time is then resumed, and the procedure repeats. Figure 6 shows the schematic procedure. The perturbed performance measure $J(\theta + \Delta\theta)$ in the computation of *gn* is then calculated along CPP. The resulting procedure to get CPP is called the extended perturbation analysis scheme. The procedure has been proven to be valid and general EPA scheme is provided within the framework of the generalized semi-Markov process model of DEDS. For more detailed discussion and experimental results, the reader is referred to [12, 13].

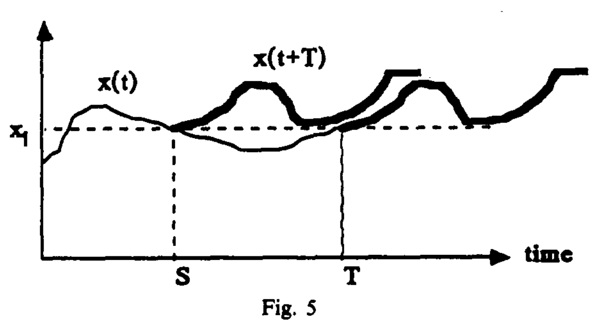

Fig. 5

†We do not consider the variance problem, which depends on the particular random seeds used [18].

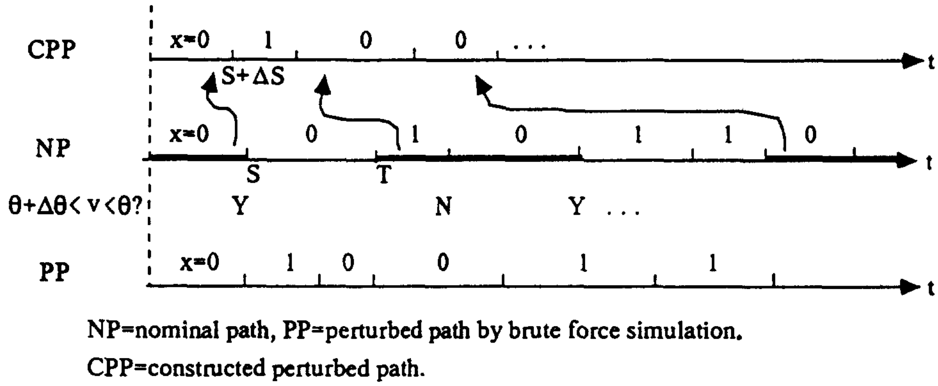

Fig. 6

4. ANALOG TO THE ACCEPTANCE–REJECTION APPROACH AND FURTHER EXTENSIONS

In this section, we discuss an interesting analog to the acceptance–rejection method of random variable generation in simulation. This analog not only gives some insight to the cut-and-paste idea but also gives rise to new algorithms.

First, let us review the acceptance–rejection method [15]. Suppose that a random variable W is distributed with density function $f_W(x)$. One scheme to generate samples of W is to generate a pair of random variable (X, Y) uniformly distributed in the area under the curve $f_W(x)$. Then it is very easy to verify that the first component X has the distribution of W. The procedure is shown in Fig. 7.

Now suppose we have this mechanism to generate samples of the random variable W, and furthermore, we want also to get samples of another random variable V with density function $f_V(v)$. The idea of acceptance–rejection is the following: If we multiply the function $f_W(x)$ by a factor C, determined by $C = \min\{c: cf_W(x) \geqslant f_V(x) \text{ for all } x\}$, then the curve $f_V(x)$ is always under $Cf_W(x)$. Hence, given the pair of random variable (X, Y) generated as in Fig. 7 for W, if (X, CY) is in the area under $f_V(x)$, then we accept X as a sample of V; otherwise reject it. This procedure is shown in Fig. 8.

The similarity of the EPA algorithm and the acceptance–rejection method is now apparent. In EPA, we cut out portions of the nominal path and accept other portions. Thus the EPA algorithm can be treated as a way of doing acceptance–rejection for stochastic process (or precisely Markov process). This topic has also been discussed in Ho *et al.* [22].

Now let us take a look at another variation based on this line of ideas. Suppose we do not start generating the sample (X, Y) for W. Instead, we generate a pair of random variables (X', Y') uniformly distributed in the area under $f_W(w)$ or $f_V(v)$, i.e. we define $A_{WV} = A_W + A_V$.† If (X', Y') is in A_W then accept X' as a sample of W, if (X', Y') is in A_V, then accept X as sample of V. This procedure is given in Fig. 9.

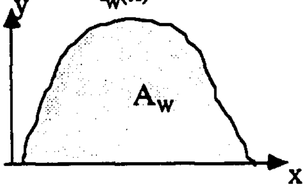

Fig. 7. Generate random variable $W \sim f_W(w)$: (i) generate $(X, Y) \sim$ uniform (A_W); (ii) let $W = X.d$

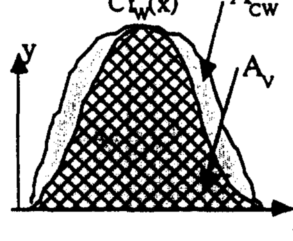

Fig. 8. Generate random variable $V \sim f_V(v)$: (i) use existing (X, Y); (ii) if (X, CY) is in A_V, then let $V = X$; otherwise, reject X.

†The operator "+" denotes the union set of the two operands.

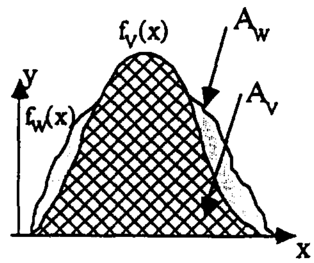

Fig. 9

Clearly in this procedure, if the two functions $f_W(w)$ and $f_V(v)$ are very similar, then a lot of times, (X', Y') will be in both area A_W and A_V. Thus the procedure is more efficient than using twice the procedure in Fig. 7. Furthermore, if we are generating many slightly different random variables, we can very efficiently explore the advantage of the last scheme. This idea has also been applied in the content of stochastic process, and the resulting procedure is given in Li [21].

5. CONCLUDING REMARKS

We show that the simple idea of invariance under cut-and-paste offers a general approach for the sample path generation of Markovian DEDS. There may be a lot more new competing algorithms in the line of this idea to be developed in the future. The acceptance–rejection analog between EPA and random variable generation gives more insight to the new algorithms.

REFERENCES

1. A. H. Levis *et al.*, Challenges to control: a collective view. *IEEE Trans. autom. Control* **AC32,** 275–285 (1987).
2. Y. C. Ho, Basic research, manufacturing automation, and putting the cart before the horse. *IEEE Trans. autom. Control* **AC32,** 1042–1043 (1987).
3. P. J. Ramadge and W. M. Wonham, Supervisory control of a class of discrete event processes. *SIAM J. control Optim.* **25,** 206–230 (1987).
4. K. Inan and P. Varaiya, Algebras of discrete event models. Submitted manuscript.
5. G. Cohen, Algebraic tools for the performance evaluation of discrete event systems. Submitted manuscript.
6. K. Matthes, Zur Theorie der Bedienungsprozesse. *Trans. 3rd Prague Conf. Information Theory* (1962).
7. Y. C. Ho, X. X. Eyler and X. X. Chien. A gradient technique for general buffer storage design in a production line. *Int. J. Prod. Res.* **17**(6), 557–580 (1979).
8. F. Baskett, K. M. Chandy, R. R. Muntz and F. G. Palacios, Open, closed and mixed networks of queues with different classes of customers. *J. ACM* **22**(2), 248–260 (1975).
9. Y. C. Ho and X. Cao, Perturbation analysis and optimization of queuing networks. *J. Optim. Theory Applic.* **40**(4) (1983).
10. Y. C. Ho, Performance evaluation and perturbation analysis of discrete event dynamic system, perspective and open problems. *IEEE Trans. autom. Control* **AC32,** 563–572 (1987).
11. Y. C. Ho, Perturbation analysis explained. *IEEE Trans. autom. Control.* In press.
12. Y. C. Ho and S. Li, Extensions of infinitesimal perturbation analysis. *IEEE Trans. autom. Control* **33,** 427–438 (1988).
13. S. Li, Extended perturbation analysis of discrete event dynamical systems. Ph.D. Thesis, Division of Applied Sciences, Harvard University (1988).
14. H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems* Springer, Berlin (1978).
15. A. Law and W. D. Kelton, *Simulation Modeling and Analysis.* McGraw-Hill, New York (1982).
16. X. Cao, Convergence of parameter sensitivity estimates in a stochastic experiment. *IEEE Trans. autom. Control* **AC30** (9) (1985).
17. R. Suri and M. Zazanis, Perturbation analysis is exact for the M/G/1 queue! *Proc. 23rd IEEE Conf. Decision and Control* (1984).
18. X. Cao, On the sample functions of queuing networks with applications to perturbation analysis. Submitted manuscript.
19. X. Cao, First order perturbation analysis of a simple multiclass finite source queue. *Perform. Eval.* (1988).
20. P. Heidelberg, X. Cao, M. Zananis and R. Suri, Convergence properties of infinitesimal perturbation analysis estimates. Submitted manuscript.
21. S. Li, Marked event method of discrete event dynamical systems. Submitted manuscript.
22. Y. C. Ho, S. Li and P. Vakili, On the efficient generation of discrete event sample paths. Submitted manuscript.